FUJI FACTS

Newsletter of the
Atari Computer Enthusiasts of Columbus

Volume 5, № 3

March, 1987

Your Artwork

Could Be Here!

(please contact the editor
for details)

This newsletter is written and published monthly by the Atari Computer Enthusiasts of Columbus (**ACEC**). ACEC is an independent, non-profit organization interested in exchanging information about any and all Atari Home Computer Systems.

Our main meetings are held on the second Monday of each month at 7:15 p.m., at DeSales High School (on Karl Road, just south of Morse Rd.), and are open to the public. Other Special Interest meetings are held as announced at the main meeting.

Dues are $12.00 per year, and entitle members to all club benefits (Newsletter, Disk of the Month, Publications Library, SIG meetings, group discounts at selected area merchants, etc.).

Fuji Facts welcomes contributions of articles, reviews, editorials and any other material relating to the Atari computers, or compatible hardware devices and software packages.

The cover of this month's newsletter was printed with a Star SG-10 dot matrix printer, using XLent Software's TypeSetter 130. The newsletter itself was printed with a Star PowerType daisywheel printer in elite pitch, using Batteries Included's PaperClip version 2.0 on a 256K RAM modified Atari 800 XL.

Our permanent mailing address is:
P.O. Box 849
Worthington, OH 43085

PRESIDENT:
Charles Lusco
4624 Channing Terrace, #C
Columbus, OH 43232
863-4016

VICE-PRESIDENT:
Dave Beck
1194 Country Club Road
Columbus, OH 43227
863-8600

PUBLICATIONS LIBRARIAN:
Bill Morgens
4943 Botsford Drive
Columbus, OH 43232
861-2431

MEMBERSHIP CHAIRMAN:
Mike Compton
1342 Gumwood Drive
Columbus, OH 43229
885-3757

NEWSLETTER EDITOR:
Warren Lieuallen
1652 Hess Boulevard
Columbus, OH 43212
488-3977

DISK LIBRARIAN:
Paul Pelfrey
2199-D Cortside Drive
Columbus, OH 43232
866-1454

TREASURER:
Dave Feeney
2665 Blue Rock Boulevard
Grove City, OH 43123
871-0524

SECRETARY:
Don Bowlin
230 Orchard Lane
Columbus, OH 43214
262-6945

The Editor's Column

The Good, The Bad and The Ugly

Well, as you can see, there are a few more surprises for you this month in Fuji Facts. Unfortunately, after what I hope has been a long string of pleasant surprises, this one is a little harder to take. Due to a combination of several factors and unforeseen coincidences, our treasury is nearly empty! While this is not a life-threatening situation for ACEC, it does mean that we're going to have to cut back on some of our services for a month or so.

The most obvious "program-cut" is what you are reading right now. While I remain very enthusiastic about our new newsletter format, and am excited about the prospects of further improvements, at present we can no longer afford the luxuries of a monthly twenty four page newsletter, offset printing and first-class mailings. So, effective immediately, and until further notice, all newsletters (in their reduced page number format) must be picked up at our monthly meetings at DeSales High. I will continue to bring back issues, so if you must miss a meeting, your newsletters will be waiting for you next month.

To our members that for one reason or another are unable to attend our Monday meetings, I apologize for this inconvenience. Limited numbers of mailings can probably be done, but this depends on you contacting me, and on the progress of our "fiscal recovery programs"! I also hope to continue having copies available at Computer Express (800 Bethel Road), as well as at our regular eight-bit SIG meetings.

Those of you who have come to depend on the quality of Fuji Facts need not feel slighted. I will do my best to maintain our usual level of quality material, although I must cut back significantly in quantity. You'll probably be seeing less from Antic On-Line, and more of the best from other users' groups around the country, not to mention the best from our own members. Now more than ever before, I truly need you to contribute to

Fuji Facts. If I can't get enough "free" material to fill eight pages, I'll be forced to reduce our newsletter to a two-page pamphlet.

This reduced format, and our temporarily reduced circulation is not without its rewards, thought. For these smaller issues, I am able to arrange <u>free</u> photocopying (for a few months, anyway). Combining that with the postage savings means that, with your help, ACEC will soon be back on its feet, and back to normal operations.

So, if there is any good news in all of this, it is that this reduction is only temporary, and quite short at that (we hope). Please help us out by following the advice that Charles Lusco gave us at the meeting — please, support your club, so that we may continue to support you, our members.

Warren Lieuallen

(cont. from page 7)

run from irritating to disasterous when they pop up. For instance: the "fill" funcition usually won't entirely fill a complex shape. Also, some functions such as "fill" and the "oval" commands have to be tried twice before they actually work. Worst, the disk I/O sometimes "goes south", leaving one with a PIC which can't be saved. Not even a directory listing can be made when this happens.

Problem 4: No provision is made for creating or saving your own shape or font files. (But Suncom plans to <u>sell</u> extra shape file disks.)

That's enough. All I'd like to say now is that this flawed product may look very good after some hardware and software bug fixes. But consider the pros and cons before paying your $80 for this intriguing product.
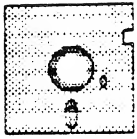
Software Review

Lords of Conquest
or How to Rule the World
With Your Atari

By Scott Spears

Electronic Arts have done it again. First with MULE, and now with the release of Lords of Conquest, they have nearly cornered the market of excellent, multi-player strategy games. Lords of Conquest is a variation on the old board game standard Risk, but with many different options.

The game allows one to four players to compete for world domination with twenty existing maps of the earth or section of the earth (USA, Europe, Australia, etc.). You also have the option of using the map-making program to build your own territories from scratch, or you can have the map maker form random countries from a set of parameters you give it. The graphics are clear and simple.

At the start of the game, you choose the type of game you will play. The game is menu driven, and shows all options you have. First, you pick the number of players -- one to four can play. All players use the same joystick which eliminates the problem that MULE had in which you had to have an Atari 400 or 800 to get the four joysticks needed to play with four players.

If one player is selected then the computer plays the opponent. With the computer as your foe you can choose a difficulty level of one through nine. Depending on the level of play selected, either you or the computer have the advantage at the start of the game.

If two or more humans play then there is no difficulty level. (All men are created equal, but the computer needs the advantage as you get more experienced.)

Next you pick the skill level. Skill level determines the number of options you have during the game. There are four skill levels, from beginner to expert. At beginners level you don´t have certain resources or boats with which to deal. At higher levels you have more resources and are allowed shipping of forces by boat.

The resources you have in the game are one of the most important elements of play. At the outset, you can set the amount of resources from low to high. Resources available are: iron, coal, timber, gold and horses. Each has its importance in the scheme of things. My advice is to concentrate on the minerals and timber. You´ll need these to build weapons and cities. If there are horses left, then grab them before taking empty territories.

The next important option is the level of chance. With this you add that random factor to the game. It affects the shipment of goods, production of resources and the outcome of battles. Production can be halted by striking workers and shipment can be stopped due to apathy. Knowing that the level of chance was high has caused me to break out into a sweat when going into battle even when I out numbered the opponent, because unless the odds are much greater in your favor, you still can lose.

You can turn the music off at the start of the game. It can be turned back on during the game.

The last option on the beginning menu scene is choosing the level of difficulty. There are nine levels from which to pick. As the level gets higher the computer gets the edge. At level one, you get to choose the first

territory, the first attack and four extra territories. On the level nine setting the computer gets those advantages.

From there, you go to map selection. As I mentioned, you can choose from a pre-made map or create your own. After a map is chosen, the resources are distributed. This can be all random or you can select the placement.

Now you are ready for territory selection. These can be the most important strategic decisions of the game. If you choose wrong, the game will be short; if you choose right then maybe you'll win. Again, my advice is go for resources first. Then, decide where you will put your stockpile. Your stockpile is the goods you produce which you will use to build cites, weapons and boats. If the territory your stockpile is in is taken, then there is looting. Your enemy will get your goods and you'll have nothing to build with next round. This usually spells defeat for you.

After carefully choosing your territories, the game is ready to begin. The turns are in year periods which are divided into three sections: development, shipment and attack.

During the development you use your resources to build weapons, cities, and boats. Each of these require different combinations of resources or you can build them with gold. After building, you place them in your territories. Each have their own offensive and defensive advantages, and you should think hard about their placement.

Next is shipment. You can ship your weapons, horses and boats. Horses and boats can haul weapons to the front lines or to reinforce unprotected backlands. Also as one of your turns during shipment you can move your stockpile.

Finally, you are prepared to attack. You get two attacks per turn. The

battles are decided on the basis of the number of supporting territories and the forces in these areas. If you outnumber the enemy you will usually win, unless you'll playing at a high element of chance. At a high level of chance, you'd better have good odds in your favor.

If all this sounds complicated, it's not. After the first time you go through it you'll understand the basics.

The documentation is clear and well written. There are some good strategy tips offered by the game designers in the back of the manual.

The game offers a save game feature, but I get so engrossed in the game that I usually play a game to its final outcome.

Conclusion:

I really enjoy Lords of Conquest. The turns move along quickly and the game play is smooth. An average game can last from thirty minutes to an hour and fifteen minutes. The computer makes a good opponent at the high levels of difficulty and less territories. With two players, the game can be nerve-racking. I must confess, I have never played with three or four players (except once when I played all four players), but there is feature which allows alliances to be formed which could add some real excitement to the game.

I have only a few complaints with the game. One is that it requires a color display to play. The territory colors all look the same in monochrome. Another fault is that you cannot use your second attack to ship items back to safer areas. And the final dig is that I don't like the title screen.

I can recommend Lords of Conquest to anyone who likes Risk. I think it is a good improvement upon that basic concept. And my final advise is protect your stockpile.
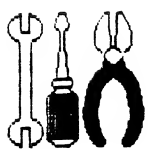
Hardware Review

The Animation Station
Reviewed by Art Canfil

I just bought an "Animation Station" (from Suncom), and have been fooling around with it this weekend. It's impressive in both it's strengths and flaws, and I'd like to share some of both with all the ACEC members.

The package retails for $79.95, and includes both hardware (a Koala-like pad), and software ("DesignLab"). There are versions of Animation Station for several popular microcomputers. This review will deal only with the Atari 8-bit version.

First, the hardware. The Animation Station pad (AS) is similar to the Koala pad or the Atari touch pad. It measures slightly larger than a Koala Pad.

The most important physical difference between the AS and the Koala is that the AS sensor pad is rectangular, like a TV or monitor screen, not square, as with the Koala's sensor pad. This means a much more realistic "feel" for freehand artwork. (It also meant that I had to readjust my coordination to compensate for a couple of years using the Koala Pad.)

The AS pad has two buttons -- the top one is called "UNDO", and the bottom is called "DO" -- and these sets of buttons are conveniently placed on both right and left sides of the pad. (Equally handy for the "Lefties" out there!)

On the back side of the pad is a handle-like extension that swings out to allow the pad to tilt up at a handy angle on a desk or table. (I personally keep the handle closed, preferring to use the AS pad in my lap.)

The AS pad also has a tiny switch at the lower-right corner, which is supposed to allow the pad to emulate "paddle, keyboard, mouse or joystick control" (according to the slickly printed box's claim). This comes perilously close to being a blatant lie: the few games using paddles might be played with the AS pad, I suppose, though rather awkwardly. But special software would have to be written for the AS pad to function as "keyboard" (huh?) or as a "joystick"! Certainly the pad does not substitute for a joystick in the existing programs I tried. (As for it emulating a mouse, I won't comment, having never had reason to use a mouse with an 8-bit Atari.)

One last functional item regarding the hardware half of the Animation Station package: when using software other than that included with the product (I used MicroIllustrator), the pad behaves as follows: Whenever the stylus is lifted, the cursor zooms to the upper-right hand corner of the screen. (With a Koala Pad, the cursor disappears when the stylus is not pressing the sensor pad surface.) Because of the way MicroIllustrator works, this makes the AS pad very awkward to use, and tends to leave nasty, unexpected lines between the last place one was drawing and the upper-right corner! This would probably be a major handicap when using many other programs, as well. Flipping the "emulator" switch back or forth seems to have no effect. (As far as my experiments have been able to determine, the emulator switch does nothing at all!)

The cream-colored case appears as tough as a Koala Pad (tough as nails), and the cord length is adequate. A nice plastic stylus is included, and the pad has a hole for its storage.

In summary of the hardware, this is a very much improved sensor pad, ergonomically speaking. But there's a

serious hardware problem in that the pad thinks it's always being pressed. And Suncom makes claims about the pad's utility which would tend to mislead most potential purchasers.

Now for the software. The included DesignLab software and data files (by Baudville, Ltd.) impressed me immediately with their power and intuitive design.

I have a copy of RAMbrandt, as well as MicroIllustrator (a.k.a. "Atari Artist", "Atari Painter" and "Koala Painter", I understand). RAMbrandt is the "power" painter's tool, but I find it buggy in some areas and so un-intuitive to use that it's a pain in the rear. (The fact that it makes files in FORTH disk format, for instance, means I spend lots of extra time laboriously translating file formats -- and sometimes making fatal mistakes.) MicroIllustrator (M-I) is highly intuitive, generally a pleasure to use. It appears to be entirely bug-free. But M-I doesn't have many "bells and whistles". Not even a "spray" brush, or the ability to move stuff around on the screen (windows and stamps). Nor does M-I have "text" functions. (You have to go into "magnify" and make your letters from individual pixels.)

But DesignLab has many "power" features, yet is very "ergonomic" and "intuitive" (techno-babble words for "easy to use").

DesignLab allows many fine aids for painting, as well as several handy shape table files which can be loaded (and the shapes then rotated and/or flipped). The shape files are especially useful for the non-artist in us all. Font files can also be loaded, and then one can hit the "text" icon, place the cursor anywhere on a PIC with the pad, and start typing from the keyboard. The software supports screen dumps using Epson, Gemini, Panasonic, or Okidata Okimate printers. The whole PIC can be scrolled in any direction on the screen, in a wrap-around manner. Available hues and luminosities

are the same as with MicroIllustrator, but mixing can be done with the four main selected colors by user choice. Windows can be defined on the screen, then moved, or even saved as window files on disk. And on, and on... And almost everything is done without having to put down the AS pad to hit the keyboard.
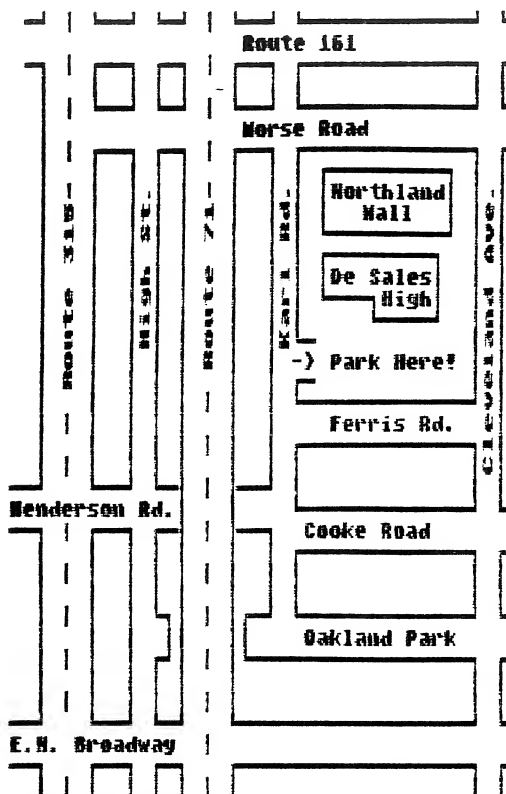
I was so impressed by the whole hardware/software combination that I immediately called several friends about my fantastic buy. Later, after two days of using the DesignLab software, my feelings are much more mixed.

Problem 1: The PICs produced by the D-L software are not compatible with either MicroIllustrator or MicroPainter. I was able to load D-L PIC files onto my screen using "SEEPIC", and dump to my Epson without trouble. Using MegaFont II, I was able to do a screen dump to my Epson, by telling M-F II that the D-L PIC was "graphics 7+/8". MicroIllustrator refused to load D-L PICs at all, claiming "file too short"(!) The only way I could "load" a D-L PIC into RAMbrandt was by telling the program that the file was in "Micro" format. But when loaded, it did strange things to RAMbrandt, and an Epson screen dump from that program gave me four disjointed parts of the PIC (some parts were missing). FADERII goes entirely insane when it tries to load a D-L PIC. RainbowDOS can load a D-L PIC as a Micro-Painter file, but trying to load one as a Koala file causes RainbowDOS to go berserk. Lastly, Digit-a-View will load the file, but produces the wrong color values.

Problem 2: All PIC files produced by D-L are 62 (single density) sectors long, regardless of the PIC's complexity. In contrast, a rather complex M-I PIC file is about half as long, and many run to less than a third of that. (I understand that M-I files are in a "compacted" form.) The large files mean expense in both floppy disks and connect time when uploading or downloading these PICs.

Problem 3: Bugs. Not many, but they

(not to scale)

Route 161

Morse Road

Route 71b | Route 71a | High St. | Route 71 | Karl Road | Cleveland Ave

Northland
Mall

De Sales
High

-) Park Here!

Ferris Rd.

Cooke Road

Oakland Park

Henderson Rd.

E. N. Broadway

An official Users' Group,
the Atari Computer
Enthusiasts of Columbus
meets on the SECOND
MONDAY of each month.
The meetings are held at
7:15 p.m., at De Sales
High School on Karl Road.
Meetings are open to the
public, and consist of
demonstrations and short
tutorials of products for
the Atari Home Computer
Systems. Dues for ACEC
are $12.00 per year, and
include a subscription to
Fuji Facts, and more!

WGL '87

Fuji Facts Newsletter
Warren Lieuallen, Editor
1652 Hess Boulevard
Columbus, OH 43212

TO:

MEETING: Apr. 13th, 7:15 pm

FUJI FACTS

Newsletter of the
Atari Computer Enthusiasts of Columbus

Volume 5, № 4

April, 1987

ATARI Word Search!

```
N E P C I T N A K C O M P U T E R P E J J A E Q R
S R B Q T S Q E T S A G E M Q M E D O M A O R L D
R N Z Q H R Y D I S K G M G S U R S V G B G D W C
A N O H C B E Y C L M W Y E K O P O A T A R I B T
E W H I O A H T U B A A M M M Y V Q T N Y B Q F Y
D Q L A T B S F T E L B A D R O F F A I Y E Y X E
N P R G A A K S J O P E K E E C J C A Q N D Q B U
R D H R T T C O E V L R E G D I R T R A C O L C C
Y L G A N U Y I V T P P I W C R K U K F A P M P F
E U E P F S R S L D T H A N D V E G U E P R Q B F
K I R H T M Z U O P M E H E T Z Z J U J H O K P Y
I N K I F E W B A J P O F I H E I N V V L G U C M
B M C C X B Z C C E O A S S R K R B V X U R U M H
H K Z S L Y E J Z V U F P R V P K H A R U A B M N
T K P C Y C V X N P Y X H C X D Q E K N Y M O L N
```

Word List:

ACEC	Graphics
Affordable	Joystick
Antic	Keyboard
Applications	Mega ST
Atari	Modem
Cartridge	Monitor
Cassette	Plotter
Computer	Pokey
Disk	Printer
Fuji	Program

INSIDE THIS ISSUE:

This newsletter is written and published monthly by the Atari Computer Enthusiasts of Columbus (**ACEC**). ACEC is an independent, non-profit organization interested in exchanging information about any and all Atari Home Computer Systems.

Our main meetings are held on the second Monday of each month at 7:15 p.m., at DeSales High School (on Karl Road, just south of Morse Rd.), and are open to the public. Other Special Interest meetings are held as announced at the main meeting.

Dues are $12.00 per year, and entitle members to all club benefits (Newsletter, Disk of the Month, Publications Library, SIG meetings, group discounts at selected area merchants, etc.).

Fuji Facts welcomes contributions of articles, reviews, editorials and any other material relating to the Atari computers, or compatible hardware devices and software packages.

The cover of this month's newsletter was printed with a Star SG-10 dot matrix printer, using XLent Software's TypeSetter 130. The newsletter itself was printed with a Star PowerType daisywheel printer in elite pitch, using Batteries Included's PaperClip version 2.0 on a 256K RAM modified Atari 800 XL.

Our permanent mailing address is:
P.O. Box 849
Worthington, OH 43085

PRESIDENT:
Charles Lusco
4624 Channing Terrace, #C
Columbus, OH 43232
863-4016

VICE-PRESIDENT:
Dave Beck
1194 Country Club Road
Columbus, OH 43227
863-8600

PUBLICATIONS LIBRARIAN:
Bill Morgens
4943 Botsford Drive
Columbus, OH 43232
861-2431

MEMBERSHIP CHAIRMAN:
Mike Compton
1342 Gumwood Drive
Columbus, OH 43229
885-3757

NEWSLETTER EDITOR:
Warren Lieuallen
1652 Hess Boulevard
Columbus, OH 43212
488-3977

DISK LIBRARIAN:
Paul Pelfrey
2199-D Cortside Drive
Columbus, OH 43232
866-1454

TREASURER:
Dave Feeney
2665 Blue Rock Boulevard
Grove City, OH 43123
371-0524

SECRETARY:
Don Bowlin
230 Orchard Lane
Columbus, OH 43214
262-6945

The Editor's Column

"That's just the way it is."
B. Hornsby

In keeping with our new, leaner look, I intend to be very brief this month (pause for the cheers and applause to die down!). In the first place, that will leave more room for all the good stuff. Secondly, I just don't have very much to say.

You all know by now that while still financially troubled (as are a surprising number of users' groups around the country), ACEC is coming back! Our recovery is slow, but steady. You can help; please, help us to help you.

This month is historic in that, for the first time since I can remember, we've got an ACEC-only issue! Every word of this month's Fuji Facts was written by one of our own. Personally, I think that's the way is should be EVERY month. Again, that's up to you.

Enough of this — let the games begin!

Warren Lieuallen
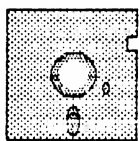
Software Review

HIT A KEY, ANY KEY
by Charles W. Brown

In past articles I have tried to explain about some of the many different memory locations within the Atari computers. Among these locations, there are some that the computer needs for its own operation; then, there are others that the programmer or the user can use for his own purposes.

Some locations contain values that never change. There are also locations that can have different values that are depended on various conditions. In this article I will try to show how one such memory location is vital to the computer and its operation.

The computer can't do anything unless it has some type of information to process. This information could be instructions as what to do (this is what a program is), or it could be some data that a program needs in order to run. Much of this information comes from the keyboard, because someone somewhere had to type the information in. Now comes the big question — How does the computer know what key is being pushed? The computer has to know so it will understand what is being fed into it. How does the computer keep track of all those keystrokes?

There is a memory location in the computer that does just that: It keeps track of what key is being pressed. This location is 764 decimal (or 2FC hex). Every time a key is pressed, it produces a certain value in this location. Each key press will create a different value. The operating system looks at this location frequently; by knowing the value, it will know what key is pressed.

Different key combinations have different values assigned to them. When you press a key by itself, it will produce a value. If you push that same key with the control key at the same time, you will get a different value. The same is true for for the shift key, and the shift and the control key as well. This way the computer will know if the shift and/or control keys have been pressed or not.

This process is done by the keyboard handler. This is a routine that is built into the operating system ROM (read only memory). Its job is to get the input from the keyboard and process it. So the first thing it has to do is know what key is being hit.

When you hit a key, the value for that key is placed into location 764. The value that is stored here is not the ATASCII or even the internal value for the character that you pressed. It is called the raw keyboard code. The value is held here until it is accepted by the keyboard handler.

Another location used by the keyboard handler is 754 (or 2F2 hex). This holds the prior keyboard character code, which is the value for the key last read and accepted by the keyboard handler.

Every time a key is accepted, a 3 is stored into location 753. This value is decremented during every stage two Vblank (every 1/60th to 1/30th a second). As you might imagine, this location is

used as counter by the keyboard handler. When a value is stored here, it just gets decremented down to 0. A zero will stay there until a new key is accepted by the keyboard handler.

Now let's look at the process for getting a key. This will be in a simplistic form. First, the keyboard handler takes the value from location 764 (the key that was just pressed). The value gets compared with the one in location 754 (the last key accepted). If the two are different, then the value in location 764 is accepted and stored into 754.

If the two values are the same, then the handler looks at the value in 753 (the counter). If the counter value is greater than 0, then the value in 764 is ignored. The computer will consider the value in 764 to be a debounce, or an echo from an old key press. If the counter in location 753 is 0, then the value in 764 will be accepted as a proper key press. This value will then be stored into location 754 in order to update it. This is why when you hold a key down, there is a slight delay before it goes into the auto-repeat function.

Then, a 255 is stored into location 764. This clears the keyboard register. A 255 tells the computer there has been no key pressed. If it didn't do that, the computer would see the same value in 764, and think that that key was being continually pressed.

I know this sounds like a complicated procedure — that's because it is! This is just a glimpse at what the computer has to do every time we hit a key.

As a programmer you can use this routine, too. Many programs use the keyboard to get the input. Let's say that you want to have a program tell the user to hit the return key. In BASIC it would look like this.

```
10 PRINT "Please hit the RETURN key."
20 IF PEEK(764)=12 THEN GOTO 50
30 IF PEEK(764)=255 GOTO 20
40 IF PEEK(764)<>12 THEN GOTO 60
50 PRINT "Thanks for hitting the return key!":POKE 764,255:GOTO 10
60 PRINT "Sorry... you hit the wrong key!":POKE 764,255:GOTO 10
```

The heart of this program are lines 20 and 30. First, you must know that the value for the return key is 12 in the keyboard code. In line 20 you are looking at location 764. This is done with the PEEK command. The second part of line 20 says that if the value is 12, then go to line 50. If instead the computer finds the value in 764 to be different, it will ignore the rest of the line. In line 30 you check to see if there has been any key pressed. This is done by seeing if a 255 is stored there. If there is no key press the program will loop back to line 20. If the value does not equal 255 then it will fall through to line 40. In this line the 764 code will be checked again. In line 50 a message tells the user they hit the return key. Then a 255 will be stored into location 764. This is done with the poke 764,255. This clears the keyboard register. Then the program will go back to line 20 to do it again. If the value found in 764 is not a 12 then program will go to line 60. For this to be true, another key had to be hit. In line 60 a message tells the user they hit the wrong key. Then, once again, it clears the keyboard register. Then the program goes back up to line 20 again.

Using the PEEK 764 method, you could use the keyboard for almost any type of users' response. Since there are 255 different key combinations, you could have 255 different choices at once. Remember that if you do get a value from the keyboard code, you have to store a 255 into that location once you're done with it. This will clear it. Otherwise, the computer will think you are still hitting the same key. This will often cause you a lot of unnecessary grief.

I hope that you understand that I have only given you a glimpse of the keyboard reading process. There are a lot more things that going on then what I have just told you about. This is just the tip of the iceberg. Anyway, I hope you have at least a little understanding on how the computer knows what key was hit, and how you can use this process for own needs. It may sound like a lot of work. But just think how would communicate with the computer if it didn't know what keys you hit?

Hardware Modification

Do-It-Yourself Surge Protection
by Kevin Watson

SURGES

Surges in power line voltages can cause massive damage in any unprotected electronic devices. A surge is simply a sudden rise in the 117 volt supply that provides power to the equipment in question. They can come from a wide variety of sources: your neighbor's arc welder, lightning in your area, a vacuum cleaner with a bad ground. The surges can range in size from a mere pop of 130 volts to a grand sizzle of 1500 volts! The damage they can cause ranges from a temporary warming of the voltage regulator that feeds your computer it's normal 5-volt diet to a blitzkrieg pulse that runs through the entire system, cooking everything it encounters before the regulator gives up the ghost. Surges are not to be confused with a similar effect called noise.

Noise is a superimposition of a high frequency signal on the normal 60 cycles per second frequency of the 117 volt power line. Although not nearly as likely to cause damage, noise can be very irritating by causing funny bands to drift across the screen of your monitor. Noise can also come from sources other than the power line, such as CB radios, etc., whereas surges (barring EMP from a nuclear explosion, God forbid!) will come from the power lines alone.

SURGE PROTECTION

The typical power line contains some protection against short circuits that occur on down the line. This is in the form of either a fuse (blows fast, must be replaced), or a circuit breaker (blows slow, is reset rather than replaced). Unfortunately, surges do their evil work and are gone in a time span of a few microseconds at the most. The only thing a fuse or circuit breaker can do is to prevent the now ruined device from starting a fire. Small consolation!

A surge protection device holds the line voltage to a pre-determined level by placing a short circuit across the line when the voltages rises above the level it is made to maintain.

Like fuses and circuit breakers there are replace-when-blown (Metal Oxide Varistors - MOV's) and reset-when-blown (gas discharge) types, the reset-when-blown ones being a little more expensive than than replace-when-blown types (until you've blown a few!). The gas-discharge types also react to surges faster than the MOV's (by the way, they reset themselves). Most commercial units use the cheaper MOV protection, which means the power strip must be repaired after the protector is tripped. The circuit descriptions provided here work for both designs, as does line-noise suppression.

The circuits are incredibly simple and MUCH cheaper to put together than their commercial counterparts (better, too if you use gas-discharge devices and noise filters).

In case the circuit is unclear from the diagram (Figure One), the surge protection devices (either MOV or gas-discharge) are placed so that one device is between each of the two regular prongs and ground, and the other one is between the two prongs. They should be placed AFTER the switch and circuit breaker (if so equipped). One set of three devices is all that is needed.

NOISE PROTECTION

The coils (L1 and L2) suppress radical changes in line frequency (noise) and should be placed AFTER the surge protection devices. No coil should be placed in the ground line, as this will impair grounding of the devices. NOTE: the coils are only rated at 2 amps. Most power strips can handle 15 amps. Each piece of equipment should have either an input amperage rating or a input watts rating printed on it somewhere. To figure amperage from a watts rating, use amps=watts/117. The amperages of devices are additive, for example: two 2-amp devices together will draw 4 amps. This example would fry the noise suppression coils.

I would suggest that you only put the coils on the outlets that would supply the piece of equipment that would have noise problems (probably just the monitor). I know, I know, every time you see an ad

for a power strip with a noise filter in it, they warn you of the "deadly danger of losing data due to line noise". The way I see it, if a computer wasn't internally protected against stray noise, it would fall prey to variations in its own clock signals!
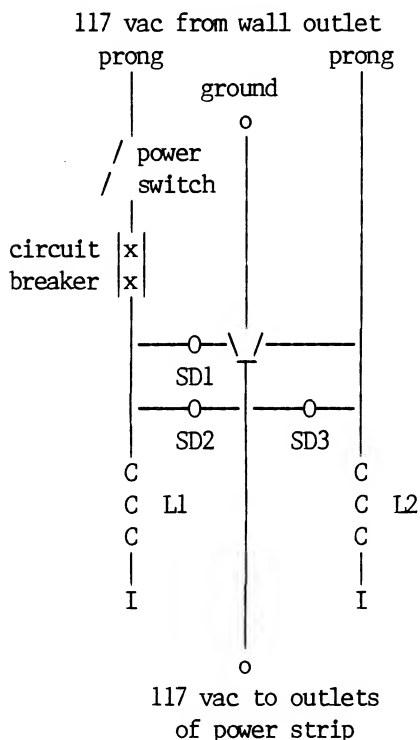
Table One: PARTS LIST
(part #'s refer to Radio Shack stock numbers)

SD1, 2, 3 (3 needed): MOV's
  Standard   # 276-570 $1.59
      or
  Heavy Duty   # 276-568 $1.69

Gas Discharge Tube # 270-811 $2.49

L1, 2 (2 needed (optional))
RF Choke   #273-102 $ .99

If you don't understand all this yet, remember that 117 volts can cook a side of beef in about a half an hour. Drop me a note on the ACEC BBS (471-8559) and I will try to help you straighten out any problems.

Figure One: Circuit Diagram

```
117 vac from wall outlet
  prong              prong
    |        ground
    |           o
   / power      |
   / switch     |
    |           |
circuit |x|     |
breaker |x|     |
    |           |
    ——o—\ /——————
       SD1   T
    ——o——|——o——
       SD2   SD3 |
    C          C
    C  L1      C  L2
    C          C
    |          |
    I          I
              |
              o
    117 vac to outlets
      of power strip
```

Software Review

THE RHYTHM COMPOSER
by Charles W. Brown

Well, I am back again trying my luck on reviewing software. I do this to let you know a little about the different programs there are so that you can make up your own mind. I try not to force my own opinions on you.

I have always enjoyed music. I would love to be able to read music and play an instrument. However, luck isn't on my side, and I never took the time to learn anything. Instead, I have purchased almost every musical program there is for the Atari, and I have played around with them all. I can assure you I am very far from creating any masterpieces. One of these days I will try and take the time; I will sit down and really try working on one and see what I can do with it. Then, watch out!

This time I will be reviewing a musical program from Antic software. It is called The Rhythm Composer. You can order it through Antic Magazine. I have found it in our local stores as well. It is a program that you can use to create different drum-type sounds. It has some very good sound effects.

The program requires 48k of RAM in your eight-bit Atari, and a disk drive. This package has 2 different programs for you to use: The Rhythm Composer, and The Sound Designer. The Rhythm Composer is used to create music with your predesigned drum sounds. The Sound Designer is used to create or edit different drum sounds.

When you first boot up the program, you will be in the rhythm composer. When you start, you will have five different drum sounds to work with. Those are the bass drum, snare drum, closed high hat cymbal, open high hat cymbal and hand clap. There can be six sounds per screen, but the opening screen only has five. So, the sixth line is a rest. There are many more rhythm sounds on the disk. You may load them in to have different sounds if you desire.

On the screen you have six different lines from top to bottom in your pattern. You have a bracket for a cursor so you can see where you are. Below the pattern there is a list of the six sounds that you are using. Each horizontal line represents one of your six sounds. If you move the cursor (with the arrow keys or a joystick) to the top line at the far left, and then hit the fire button or the return key, your first note will come from sound one; if you put the cursor on the fourth line and hit the return key, you would get a note made by your fourth sound. By moving the cursor up and down to different lines, your song will have different beats and sounds.

You can change the length of your pattern from full, to shuffle modes for waltzes, to half mode. You can change the tempo by pressing the SELECT or shift-SELECT keys. This gives you better control of your song.

I had mentioned earlier that you can use different sounds than the ones that the program started with. You just push the G key. A prompt will ask you for the name of the sound file you want. You type in the filename for the desired sound. Then, you will be asked which sound line you want the new sound to be loaded into. You just type in your choice of 1-6 to tell which line you want the new sound to go into and it will be loaded in. There are several other features that you can use with the rhythm composer. In fact, there are too many to mention here.

The other part of this package is the Sound Designer program. You can use it to edit predefined sounds, or you can create your own custom made sounds. The commands are quite simple to use — you just type in the first letter of the desired command to use it. As in the rhythm composer program you use G for Get file. This loads in a sound file for you to examine and edit. You type in a K for Keep to save a file to disk. If you want to Hear a sound you just type in a H.

The actual editing commands are very easy to use also. With them you can edit the coarse pitch, the fine pitch, the voice, the volume and sound filter. The voice part is be the most important one. It is used just like the sound command in BASIC. You just move the cursor up and down the square and pick which voice you want. As in BASIC, a ten is a pure or normal tone, while the other numbers represent different distortions of that sound. This is where

you can generate many different sound effects. In fact, you don't have to have just one voice for your whole sound. You can have your sound go from one voice to another, as many times as you want.

The filter part is a very interesting part of this program. This part controls the "AUDCTL" register. It also controls the internal clock of your computer, which controls the pitch of the sound you are hearing. Under normal conditions the clock runs at 64KHz. This part of the program will allow you to either speed up or slow down the clock, which will of course change your sound around. You can use these different sound features to alter your different rhythm sounds. This way you can change the sounds around to suit your own taste. Then, you can save them off to the disk. That way you can use your own custom sounds in the Rhythm Composer program to create your own beats.

There is a separate program on the main disk. It is a BASIC program that will print your song pattern on any printer that has 40 columns or more. The data that you wish to print must have been saved to a disk file. Then you load in this separate BASIC program and follow the instructions.
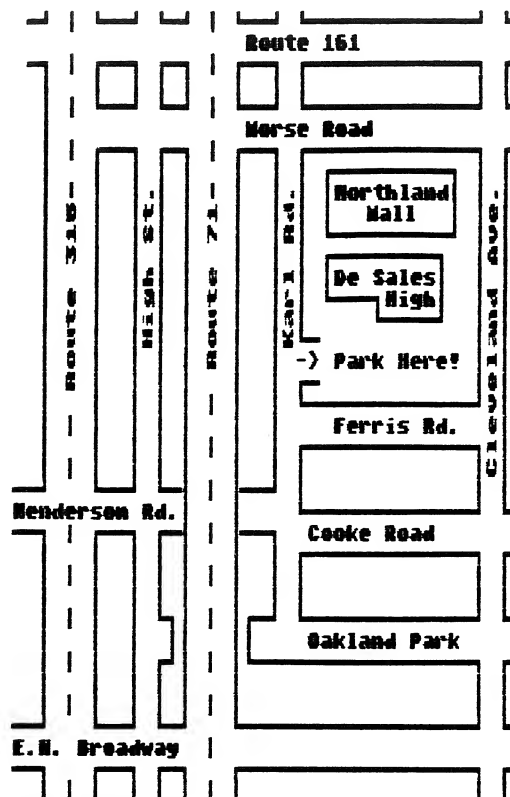
I have tried to give you a brief glimpse of what this program can do. If you are willing to work with it, you will be able to create your very own rhythm patterns to listen to. I feel that it is a good program to show off the sound capabilities of your Atari. If you like to make different sounds and musical patterns, then there is a good chance you will enjoy this program.

================

Dare to be Bare!
from the OSS Newsletter

You may recall our announcement of BareWare in the last OSS Newsletter (reprinted in the November, 1986 issue of the ACEC Newsletter - Ed.). As you may recall, we mentioned that we would be glad to review any of your own software that you thought would fit the BareWare idea. The response? Zilch. Does that mean that not a single one of you has ever written a useful program? We don't believe that for a moment, so we think that modesty or fear of rejection must be holding you back.

So, until some of you decide that your stuff is useful, BareWare products will remain restricted to stuff written by OSS associates and employees.

[not to scale]

Route 161

Morse Road

Route 315 | High St. | Route 71 | Karl Rd. | Cleveland Ave.

Northland Mall

De Sales High

-> Park Here!

Ferris Rd.

Henderson Rd.

Cooke Road

Oakland Park

E.N. Broadway

An official Users' Group, the Atari Computer Enthusiasts of Columbus meets on the **SECOND MONDAY** of each month. The meetings are held at 7:15 p.m., at De Sales High School on Karl Road. Meetings are open to the public, and consist of demonstrations and short tutorials of products for the Atari Home Computer Systems. Dues for ACEC are $12.00 per year, and include a subscription to Fuji Facts, and more!

RGL '87

TO:

MEETING: May 11th, 7:15 pm